## REMARKS

On October 7, 2008, the Examiner, and the undersigned attorney had a telephone interview during which they discussed claim 1, the Hamada reference, and the Kalkunte reference. The content of the discussion is contained herein.

103(a) Rejection Based on Chandrasekaran, Hamada and Kalkunte

Claims 1, 3, 7-12, 16-17, 20-25 and 28-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandrasekaran in view of Hamada, further in view of Kalkunte. Applicant has amended Claim 1, which now recites:

A method of handling a message received at a messaging system server, the method comprising:

> storing, in non-persistent storage, the message;
> determining whether the message has been delivered;
> if the message has been delivered, removing the message from the non-persistent storage; and
> after a configurable delay interval has elapsed and if the message has not been removed from the non-persistent storage, saving the message to persistent storage so that the message can be retrieved and delivered.

A message is stored in non-persistent storage. A determination is made regarding whether the message has been delivered. If the message has been delivered, the message is removed from the non-persistent storage. If the message has not been delivered after a configurable delay time period, the message is saved to persistent storage. Again, the delay interval is configurable.

Initially storing a message in non-persistent storage enables rapid access to the message for retransmission during the delay interval. This enables message retransmission without introducing additional latency caused by retrieving the message from persistent storage. Since initially storing the message in non-persistent storage reduces the time needed to access the message, overall system performance is increased. Saving the message to persistent storage after the delay interval has elapsed beneficially frees space in the non-persistent storage for recent messages, which increases system performance while also providing continued access to older messages from the persistent storage for later retrieval and delivery. Additionally, storing the message to persistent storage beneficially ensures that the message will not be lost in the event of network or power failure and makes the message accessible for subsequent analysis or delivery attempts.

Chandrasekaran, Hamada or Kalkunte does not disclose, teach or suggest the claimed element "if the message has been delivered, removing the message from the non-persistent storage." The Examiner cites Chandrasekaran at 12:24-12:31 for disclosing "if the message has been delivered, removing the message from the non-persistent storage." Chandrasekaran, however, does not disclose the above mentioned limitation.

Chandrasekaran discusses a system for propagating (transmitting) a message from a source site 200 to a destination site 202 (abstract). The source site tracks messages using a propagation queue 204 and a propagation table 212 (7:61-63; FIG. 2A). The propagation queue, which is in volatile memory, stores the message that is awaiting transmission (7:63-65). The propagation table, which is in non-volatile memory, does not store the message itself. The propagation table instead stores "a sequence number attribute 228, a UID attribute 230 and a state attribute 232." (FIG. 2A, 9:5-9:7).

In Chandrasekaran, at the source site: First, the message is "dequeued" (removed) from the propagation queue (7:17-19). Next, the message is assigned a propagation sequence number (7:19-21). The message is then transmitted to destination site (7:28-30). The propagation sequence number, the UID, and an initial propagation state are then stored in the non volatile memory in propagation table at the source site (7:30-32). The source site then sends a commit request to the destination site (Fig. 3, 11:23-11:25). Upon receiving a commit reply from the destination site, the source site updates the propagation state information in its non-volatile memory (11:33-11:37).

At the destination site, the received message is "enqueued" (added) in receive message queue (12:4-12:5). The destination site then waits for the commit request from the source site. If the destination site does not get a commit request and the destination site determines that the source site has failed, the destination site dequeues the message from the receive message queue (12:24-12:31). Otherwise, upon receiving the commit request from the source site, "the received message data is stored into non-volatile memory at destination site." (12:33-12:35). The destination site then sends a commit reply to indicate that the changes contained in the received message have been committed (12:47-12:50).

The Examiner cites Chandrasekaran at 12:24-12:31 as disclosing "if the message has been delivered, removing the message from the non-persistent storage." Again, Chandrasekaran at 12:24-12:31 discloses removing the message from receive message queue if the source site has failed and not if the message has been delivered. Accordingly, the requirement in Chandrasekaran for removing the message is completely different from the requirement for removing the message in claimed invention. Chandrasekaran therefore does not disclose, teach, or suggest "if the message has been delivered, removing the message from the non-persistent storage."

Chandrasekaran also does not disclose, teach or suggest "after a configurable delay interval has elapsed and if the message has not been removed from the non-persistent storage, **saving the message to persistent storage so that the message can be retrieved and delivered**." (emphasis added). The Examiner cites Chandrasekaran at 7:30-39 as disclosing the above mentioned limitation. Chandrasekaran, however, at 7:30-39 discloses saving "propagated message data," and not the "message," in non-volatile memory.

Chandrasekaran clearly states that the "propagated message data" is not the same as "message." Chandrasekaran defines "message" as a "combination of the message data and its associated header information." (7:11-7:13). Chandrasekaran then defines "propagated message data" as "propagation sequence number, the UID, and an initial propagation state." (7:30-7:34). Chandrasekaran therefore does not disclose, teach or suggest "saving the message to persistent storage so that the message can be retrieved and delivered."

Assuming that the Examiner interprets "propagated message data" in Chandrasekaran as "message" in the invention, Chandrasekaran still does not disclose the above mentioned limitation. Chandrasekaran stores propagated message data in non-volatile memory because the storage "allows the source site to determine, even after a source site failure, whether a particular message has previously been propagated to the destination site." (7:34-7:38). The source site does not store the propagated message data so that the source site can retrieve and deliver the propagated message data to the destination site. This is because the source site uses propagated message data for internal housekeeping and not for delivering the propagated message data to external entities like destination site. The source site therefore does not "retrieve[] and deliver[]" propagated message data. In other words, the source site does not store the propagated message data "to persistent storage so that the [propagated message data] can be retrieved and delivered." Accordingly, Chandrasekaran does not disclose, teach, or suggest "saving the message to persistent storage so that the message can be retrieved and delivered."

The Examiner next sites Hamada at Fig. 21, Fig. 23, and 17:35-17:65, as disclosing "after a configurable delay interval has elapsed and **if the message has not been removed from the non-persistent storage, saving the message to persistent storage** so that the message can be retrieved and delivered." (emphasis added). Hamada, however, does not disclose this limitation.

Hamada discusses a "redundant message processing system." (abstract). In the Hamada system, both the transmitter and the receiver side have a current system and a stand-by system (Fig 21, 17:15-17:21). Both the current system and the stand-by system share a common non-volatile memory (Fig. 19, 15:50-15:52, Fig. 21, 17:25-17:28). The current system stores the content of the transmitted or received message to this common non-volatile memory (17:25-17:28). Now, if the current transmitter system goes down, the stand-by transmitter system reads the transmitted message from the common non-volatile memory and retransmits the message to the receiver (Fig. 22, 17:66-18:12). In this manner, the Hamada system delivers a redundant message processing system.

To make sure that the redundant or stand-by system is able to re-transmit the message, the Hamada system stores every message in non-volatile memory before transmitting the message. In Hamada system, "the content of message ID and the content of message are … stored in the non-volatile memory … [and] then a request is made … for a message transmission." (17:35-17:65). This storage of every message in non-volatile memory is critical for the Hamada redundant message processing system. If Hamada's current system does not store all messages in non-volatile memory, the stand-by system would not be able to redeliver already transmitted messages because the messages may not exist in non-volatile memory. The Hamada system therefore would not be functional without storing every message in non-volatile memory. Thus, the Hamada system teaches away from storing only those messages that have "not been removed from non-persistent storage … after a configurable delay interval has elapsed." Accordingly, Hamada does not disclose, teach or suggest "after a configurable delay interval has elapsed and if the message has not been removed from the non-persistent storage, saving the message to persistent storage so that the message can be retrieved and delivered."

The Examiner next cites Kalkunte at 5:15-5:44 and 6:65-6:67 as disclosing "after a **configurable delay** interval has elapsed and if the message has not been removed from the non-persistent storage, saving the message to persistent storage so that the message can be retrieved and delivered." (emphasis added). Kalkunte, however, uses the time delay for a completely different purpose. Kalkunte uses time delay to restart the server and clients in a cluster at varying time instances (abstract). Additionally, the time delay is randomized and not configurable (abstract, 6:60-7:1). Because the time delay is random, Kalkunte does not disclose a time delay that can be configured. Kalkunte therefore does not disclose, teach, or suggest "after a **configurable delay** interval has elapsed and if the message has not been removed from the non-persistent storage, saving the message to persistent storage so that the message can be retrieved and delivered."

In sum, Chandrasekaran, Hamada and Kalkunte, individually or in combination, do not disclose, teach or suggest "if the message has been delivered, removing the message from the non-persistent storage" or "after a configurable delay interval has elapsed and if the message has not been removed from the non-persistent storage, saving the message to persistent storage so that the message can be retrieved and delivered."

Therefore, claim 1 (as amended) is patentable over Chandrasekaran, Hamada, and Kalkunte alone and in combination. Independent claims 16 and 24 (as amended) recite similar language and are also patentable over Chandrasekaran, Hamada, and Kalkunte alone and in combination, for at least the same reasons.

<u>103(a) Rejection Based on Chandrasekaran, Hamada, Kalkunte, and Stein</u>

Claims 4-6, 13-15, 18-19, and 26-27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Chandrasekaran in view of Hamada, further in view of Kalkunte, further in view of Stein. Applicant respectfully traverses.

Independent claim 13 recites “dynamically determining a delay time period” and “after the determined delay period has elapsed and if the message has not been removed from the non-persistent storage, saving the guaranteed message to persistent storage so that the guaranteed message can be retrieved and delivered.” Since that language is similar to the language of claim 1, the arguments above are hereby incorporated to apply to amended claim 13.

As explained above, Chandrasekaran, Hamada and Kalkunte do not disclose, teach or suggest the claimed element “if the guaranteed message has been delivered, removing the message from the non-persistent storage” or “after the determined delay period has elapsed and if the message has not been removed from the non-persistent storage, saving the guaranteed message to persistent storage so that the guaranteed message can be retrieved and delivered.”

Stein does not remedy this deficiency. Stein discusses enabling clients to perform electronic mail services when the network is unavailable (abstract). At most, Stein discloses pre-loading channel resources into a memory, specifically a persistent storage, during electronic mail channel processing by a mobile device (7:51-58). Stein saves all resources to persistent storage so that the mobile device can subsequently access the resources regardless of network availability. Thus, Stein saves all channel resources to persistent storage before they are transmitted. In contrast, the claimed invention does not save a message into persistent storage until after a delay interval has elapsed.

Thus, Stein does not disclose, teach, or suggest the claimed element "after the determined delay period has elapsed and if the message has not been removed from the non-persistent storage, saving the guaranteed message to persistent storage so that the guaranteed message can be retrieved and delivered."

Therefore, claim 13 is patentable over Chandrasekaran, Hamada, Kalkunte and Stein alone and in combination.

The claims not specifically mentioned above depend from claims 1, 13, 16 or 24 (directly or indirectly), which were shown to be patentable over Chandrasekaran, Hamada, and Kalkunte (claims 1, 16, and 24) or Chandrasekaran, Hamada, Kalkunte and Stein (claim 13), both individually and in combination. In addition, these claims recite other features not included in claims 1, 13, 16, or 24. Thus, these claims are patentable over Chandrasekaran Hamada, and Kalkunte (claims 3, 7-12, 17, 20-23, 25, and 28-31) or Chandrasekaran, Hamada, Kalkunte, and Stein (claims 4-6, 14-15, 18-19, and 26-27), both individually and in combination, for at least the reasons discussed above, as well as for the elements that they individually recite.

Applicant respectfully submits that the pending claims are now allowable over the cited art of record and requests that the Examiner allow this case. The Examiner is invited to contact the under signed in order to advance the prosecution of this case.

Respectfully Submitted,
WILLIAM M. CULLEN ET AL

Dated: October 21, 2008

By: /Greg T. Sueoka/
Greg T. Sueoka, Reg. No.: 33,800
Attorney for Applicants
Fenwick & West LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94041
Tel.: (650) 335-7194
Fax.: (650) 938-5200